

# VB.NET - MODIFIERS

The modifiers are keywords added with any programming element, to give some especial emphasis on how the programming element will behave, or will be accessed in the program

For example, the access modifiers: Public, Private, Protected, Friend, Protected Friend etc., indicates the access level of a programming element like a variable, constant, enumeration or a class.

## List of Available Modifiers in VB.Net

The following table provides the complete list of VB.Net modifiers:

S.N	Modifier	Description
1	Ansi	Specifies that Visual Basic should marshal all strings to American National Standards Institute (ANSI) values regardless of the name of the external procedure being declared.
2	Assembly	Specifies that an attribute at the beginning of a source file applies to the entire assembly.
3	Async	Indicates that the method or lambda expression that it modifies is asynchronous. Such methods are referred to as async methods. The caller of an async method can resume its work without waiting for the async method to finish..
4	Auto	The <i>charsetmodifier</i> part in the <b>Declare</b> statement supplies the character set information for marshaling strings during a call to the external procedure. It also affects how Visual Basic searches the external file for the external procedure name. The Auto modifier specifies that Visual Basic should marshal strings according to .NET Framework rules.
5	ByRef	Specifies that an argument is passed by reference, i.e., the called procedure can change the value of a variable underlying the argument in the calling code. It is used under the contexts of: <ul style="list-style-type: none"><li>• Declare Statement</li><li>• Function Statement</li><li>• Sub Statement</li></ul>
6	ByVal	Specifies that an argument is passed in such a way that the called procedure or property cannot change the value of a variable underlying the argument in the calling code. It is used under the contexts of: <ul style="list-style-type: none"><li>• Declare Statement</li><li>• Function Statement</li><li>• Operator Statement</li><li>• Property Statement</li></ul>

		<ul style="list-style-type: none"> <li>• Sub Statement</li> </ul>
7	Default	Identifies a property as the default property of its class, structure, or interface.
8	Friend	<p>Specifies that one or more declared programming elements are accessible from within the assembly that contains their declaration, not only by the component that declares them.</p> <p><b>Friend</b> access is often the preferred level for an application's programming elements, and Friend is the default access level of an interface, a module, a class, or a structure.</p>
9	In	It is used in generic interfaces and delegates.
10	Iterator	Specifies that a function or Get accessor is an iterator. An <b>iterator</b> performs a custom iteration over a collection.
11	Key	The Key keyword enables you to specify behavior for properties of anonymous types.
12	Module	Specifies that an attribute at the beginning of a source file applies to the current assembly module. It is not same as the Module statement.
13	MustInherit	Specifies that a class can be used only as a base class and that you cannot create an object directly from it.
14	MustOverride	Specifies that a property or procedure is not implemented in this class and must be overridden in a derived class before it can be used.
15	Narrowing	Indicates that a conversion operator (CType) converts a class or structure to a type that might not be able to hold some of the possible values of the original class or structure.
16	NotInheritable	Specifies that a class cannot be used as a base class.
17	NotOverridable	Specifies that a property or procedure cannot be overridden in a derived class.
18	Optional	Specifies that a procedure argument can be omitted when the procedure is called.
19	Out	For generic type parameters, the Out keyword specifies that the type is covariant.
20	Overloads	Specifies that a property or procedure redeclares one or more existing properties or procedures with the same name.
21	Overridable	Specifies that a property or procedure can be overridden by an identically named property or procedure in a derived class.
22	Overrides	Specifies that a property or procedure overrides an identically named property or procedure inherited from a base class.
23	ParamArray	ParamArray allows you to pass an arbitrary number of arguments to the procedure. A ParamArray parameter is always declared using ByVal.
24	Partial	Indicates that a class or structure declaration is a partial definition of the class or structure.

25	Private	Specifies that one or more declared programming elements are accessible only from within their declaration context, including from within any contained types.
26	Protected	Specifies that one or more declared programming elements are accessible only from within their own class or from a derived class.
27	Public	Specifies that one or more declared programming elements have no access restrictions.
28	ReadOnly	Specifies that a variable or property can be read but not written.
29	Shadows	Specifies that a declared programming element redeclares and hides an identically named element, or set of overloaded elements, in a base class.
30	Shared	Specifies that one or more declared programming elements are associated with a class or structure at large, and not with a specific instance of the class or structure.
31	Static	Specifies that one or more declared local variables are to continue to exist and retain their latest values after termination of the procedure in which they are declared.
32	Unicode	Specifies that Visual Basic should marshal all strings to Unicode values regardless of the name of the external procedure being declared.
33	Widening	Indicates that a conversion operator (CType) converts a class or structure to a type that can hold all possible values of the original class or structure.
34	WithEvents	Specifies that one or more declared member variables refer to an instance of a class that can raise events.
35	WriteOnly	Specifies that a property can be written but not read.